02

Creacion metodo

Transcripción

[00:00] Sean bienvenidos a la tercera clase del curso de introducción a objetos. Vamos a comenzar ya a profundizar en conceptos un poco más avanzados en lo que se refiere a este paradigma de orientación a objetos, descubriendo cada vez más por qué es tan útil y qué ventajas nosotros podemos obtener de esto.

[00:19] Vamos a ver un poco ahora cómo implementar ciertos comportamientos, ciertos métodos a nuestros objetos ya creados. Entonces, para retomar lo que estábamos haciendo en la clase anterior, vamos a volver a nuestro objeto cuenta. Ya habíamos definido que nuestra cuenta va a ser digamos un objeto creado por cuatro campos: saldo, agencia, número y título, pero que a la vez nosotros no queremos solamente un repositorio de datos.

[00:52] Nosotros vamos a implementar cierto comportamiento en este objeto. Entonces, ¿qué comportamiento yo podría esperar de una típica cuenta bancaria? Las operaciones de depositar, retirar dinero y transferir. Quizás puede que se les ocurren mucho más métodos para una cuenta bancaria, de hecho hay una infinidad, pero para fines didácticos vamos a considerar solamente estos tres.

[01:19] Por lo tanto, solamente para resumir aquí, nuestra cuenta va a tener nuestros campos saldo, agencia número y titular. Y a la vez vamos a implementar ciertos pasos, ciertos comportamientos para que nuestra cuenta pueda depositar, o mejor dicho, para que podamos depositar dinero a nuestra cuenta, para que podamos retirar dinero de nuestra cuenta y para transferir dinero de nuestra cuenta a otra cuenta.

[01:49] ¿Cómo lo hacemos en código Java? Nuevamente vamos a nuestro Eclipse. Tenemos aquí la clase anterior que fue crear cuenta y vamos a crear otra clase, que en este caso va a ser prueba métodos. Vamos aquí y vamos a dejar esto así por el momento: vamos a dar una mirada rápida a nuestra clase cuenta, a nuestra entidad cuenta.

[02:26] Entonces, sabemos que cada uno de estos campos aquí son los atributos que tiene nuestra cuenta: saldo, agencia, número, titular. ¿Cómo yo le puedo decir a mi cuenta que ahora yo quiero depositarle dinero a mi cuenta? Por ejemplo, hasta este momento, si yo quería, vamos a hacer aquí, creo que era aquí. Sí, perfecto.

[02:57] Si yo quería aumentar mi saldo, yo hacía simplemente una instancia de cuenta, creaba una nueva cuenta, e igualaba el saldo con la cantidad que yo quería. Es una forma de depositar dinero, sí, porque de hecho cada vez que yo deposito dinero a mi cuenta, yo aumento mi saldo, entonces como primera premisa, ¿qué tenemos?

[03:22] Para depositar dinero a mi cuenta necesito que mi cuenta incremente el saldo. Volvemos aquí a nuestra clase cuenta y vamos ya a definir nuestro primer método. En Java, o mejor dicho en la programación orientada a objetos, definimos como método toda secuencia de pasos para realizar alguna cosa o alguna acción.

[03:46] en este caso nuestro método va a ser depositar la cuenta, entonces vamos a nombrarlo como depositar, perfecto. Y como todo método, puede o no puede recibir parámetros. ¿Qué son los parámetros? Parámetros es la información necesaria que ese método necesitaría usar. Digamos que es el input que le vamos a dar al método. Hay métodos que no usan parámetros, que en este caso sería solamente especificar unos paréntesis vacíos, y también hay métodos que necesitan paréntesis.

[04:24] En este caso, nosotros para depositar dinero a una cuenta, ¿qué necesitamos? Necesitamos especificar cuánto dinero queremos depositar. Para eso vamos a definir que el valor va a ser nuestro nombre de variable que va a representar la cantidad de dinero que queremos depositar en nuestra cuenta. Y al igual que todo en Java, como todo lenguaje fuertemente tipado, necesitamos definir qué tipo de variable es valor.

[04:55] Como estamos trabajando con nuestro saldo en double, vamos a asignarle un double. Perfecto. Sigue aún en color rojo, aún no terminamos la declaración. ¿Por qué? Porque al igual que digamos los iteradores como for, while, o la sentencia if else, el método también tiene un alcance, también tiene un scope, y ya aprendimos que el scope lo definimos con nuestras llaves.

[05:32] Así nosotros tenemos ya cerrado nuestro alcance del método. ¿Esto por qué es? Porque en Java tenemos dos tipos de método. Tenemos los métodos que retornan valor y tenemos los métodos que no retornan valor. En este caso, por fines didácticos, solamente como introducción vamos a usar un método que no retorne valor, digamos que solamente él va a ejecutar una serie de cosas y no nos va a decir nada.

[05:59] Solamente va a hacer lo que tiene que hacer y no nos va a retornar absolutamente nada. Ese tipo de método usa la palabra reservada void. Con void, vemos que el código ya está compilando. A pesar de que nosotros tenemos el método completamente vacío, ya está compilando correctamente. ¿Por qué?

[06:23] Porque void no espera retornar nada, de hecho él ni siquiera espera que se ejecute algo aquí adentro. Él nos está diciendo: "Yo voy a ejecutar esto y no voy a retornar nada aquí afuera". Ya con el método definido, ahora tenemos que llamarlo. Para eso vamos a volver a nuestra clase prueba métodos y vamos a crear nuestro ya conocido método main.

[06:50] Yo creo que ahora ya está comenzando a quedar un poco más claro a qué hacemos referencia con main, y ya vemos que entendemos la primera palabra reservada, en este caso es void. Nuevamente void es un método que no retorna nada. Ahora, a modo de recordar nuestra clase anterior, vamos a crear una nueva instancia de cuenta.

[07:11] Para esto llamamos la clase cuenta, y si apretamos "Ctrl + espacio" nos vamos a dar cuenta que Eclipse automáticamente nos va a autocompletar el nombre de la variable con el mismo nombre del objeto. Esta es una práctica muy común en Java. Solemos llamar a la variable y de igual forma que en la cuenta, siempre que sea usado para algo genérico.

[07:41] Y ya esto lo igualamos a new, "Ctrl + espacio", cuenta, que ya nos está sugiriendo ahí. Enter, paréntesis, punto y coma y lo tenemos listo. Ahora, cuenta puede hacer referencia a cualquier cuenta. Como les dije, si va a ser una variable que no vamos a reutilizar mucho, o mejor dicho que hace referencia a algo muy genérico, tiene sentido sí llamarla igual que el objeto. Pero en este caso yo quiero que esta sea la cuenta de Diego, mi cuenta.

[08:19] Entonces, tranquilamente yo puedo llamarla cuentaDeDiego y va a compilar sin problemas porque es el nombre de mi variable. Podría llamarla también miCuenta y no hay ningún problema. ¿Por qué? Porque es el nombre de mi variable. Ahora, sabiendo que mi cuenta hace referencia a un objeto cuenta, si yo quisiera incrementar mi saldo, yo tengo dos opciones: o incrementarlo directamente o llamar al método depositar.

[08:54]Vamos a hacerlo como ya lo hemos hecho en la clase anterior, que sería simplemente miCuenta.saldo = 300. Perfecto. Ahora, yo aquí ya incrementé 300 soles en este caso, que es la moneda de Perú. Vamos a usarla como ejemplo. Yo aumenté 300 soles a mi saldo. Ahora,. Al igual que yo he llamado saldo aquí, como parte, como atributo de cuenta, en el caso de los métodos, de igual forma vemos que depositar también es algo que le pertenece al objeto cuenta.

[09:40] Por lo tanto, si yo deseo llamar ese método, sería tanto como decir miCuenta, punto, y vemos que aquí abajo él ya nos está llamando al método depositar. Él ya identificó que depositar también le pertenece a nuestro objeto cuenta. Le podemos dar doble clic, automáticamente Eclipse nos dice: "Okay, tú necesitas especificar un parámetro que en este caso es el tipo double y es el valor de la cuenta".